fc	fractional capability variable
x, g, a, b	expression variable
k	integer variable
el	array-element variable
ℓ	location variable

f	::=			fractional capability
	$\mid$	fc		variable
	$\mid$	$\mathbf{Z}$		zero
	$\mid$	$\mathbf{S}\, f$		successor
t	::=			linear type
	$\mid$	$\mathbf{unit}$		unit
	$\mid$	$\mathbf{bool}$		boolean (true/false)
	$\mid$	$\mathbf{int}$		63-bit integers
	$\mid$	$\mathbf{elt}$		array element
	$\mid$	$f\ \mathbf{arr}$		arrays
	$\mid$	$f\ \mathbf{mat}$		matrices
	$\mid$	$!t$		multiple-use type
	$\mid$	$\forall\, fc.t$	$\mathsf{bind}\ fc\ \mathsf{in}\ t$	frac. cap. generalisation
	$\mid$	$t \otimes t'$		pair
	$\mid$	$t \multimap t'$		linear function
	$\mid$	(t)	S	parentheses
v	::=			values
	$\mid$	p		primitives
	$\mid$	x		variable
	$\mid$	$()$		unit introduction
	$\mid$	$\mathbf{true}$		true
	$\mid$	$\mathbf{false}$		false
	$\mid$	k		integer
	$\mid$	ℓ		heap location
	$\mid$	el		array element
	$\mid$	$\mathbf{Many}\ v$		!-introduction
	$\mid$	$\mathbf{fun}\, fc \to v$		frac. cap. abstraction
	$\mid$	$v[f]$		frac. cap. specialisation
	$\mid$	(v, v')		pair introduction
	$\mid$	$\mathbf{fun}\, x : t \to e$	$\mathsf{bind}\ x\ \mathsf{in}\ e$	abstraction
	$\mid$	$\mathbf{fix}\, (g, x : t, e : t')$	$\mathsf{bind}\ g \cup x\ \mathsf{in}\ e$	fixpoint
e	::=			expression
	$\mid$	p		primitives
	$\mid$	x		variable
	$\mid$	$\mathbf{let}\, x = e\, \mathbf{in}\, e'$	$\mathsf{bind}\ x\ \mathsf{in}\ e'$	let binding
	$\mid$	$()$		unit introduction
	$\mid$	$\mathbf{let}\, () = e\, \mathbf{in}\, e'$		unit elimination
	$\mid$	$\mathbf{true}$		true
	$\mid$	$\mathbf{false}$		false
	$\mid$	$\mathbf{if}\, e\, \mathbf{then}\, e_1\, \mathbf{else}\, e_2$		if
	$\mid$	k		integer
	$\mid$	ℓ		heap location
	$\mid$	el		array element
	$\mid$	$\mathbf{Many}\ e$		!-introduction

	$\mid$	$\mathbf{let\,Many}\ x = e\ \mathbf{in}\ e'$	!-elimination
	$\mid$	$\mathbf{fun}\, fc \to e$	frac. cap. abstraction
	$\mid$	$e[f]$	frac. cap. specialisation
	$\mid$	(e, e')	pair introduction
	$\mid$	$\mathbf{let}\,(a, b) = e\,\mathbf{in}\,e'$ bind $a \cup b$ in e'	pair elimination
	$\mid$	$\mathbf{fun}\, x : t \to e$ bind x in e	abstraction
	$\mid$	$e\ e'$	application
	$\mid$	$\mathbf{fix}\,(g, x : t, e : t')$ bind $g \cup x$ in e	fixpoint

C	::=		evaluation contexts
	$\mid$	$\mathbf{let}\, x = [-]\,\mathbf{in}\, e$ bind x in e	let binding
	$\mid$	$\mathbf{let}\,() = [-]\,\mathbf{in}\, e$	unit elimination
	$\mid$	$\mathbf{if}\,[-]\,\mathbf{then}\, e_1\,\mathbf{else}\, e_2$	if
	$\mid$	$\mathbf{Many}\,[-]$	!-introduction
	$\mid$	$\mathbf{let\,Many}\ x = [-]\,\mathbf{in}\, e$	!-elimination
	$\mid$	$\mathbf{fun}\, fc \to [-]$	frac. cap. abstraction
	$\mid$	$[-][f]$	frac. cap. specialisation
	$\mid$	$([-], e)$	pair introduction
	$\mid$	$(v, [-])$	pair introduction
	$\mid$	$\mathbf{let}\,(a, b) = [-]\,\mathbf{in}\, e$ bind $a \cup b$ in e	pair elimination
	$\mid$	$[-]e$	application
	$\mid$	$v[-]$	application

p	::=		primitive
	$\mid$	$\mathbf{not}$	boolean negation
	$\mid$	$(+)$	integer addition
	$\mid$	$(-)$	integer subtraction
	$\mid$	$(*)$	integer multiplication
	$\mid$	$(/)$	integer division
	$\mid$	$(=)$	integer equality
	$\mid$	$(<)$	integer less-than
	$\mid$	$(+.)$	element addition
	$\mid$	$(-.)$	element subtraction
	$\mid$	$(*.)$	element multiplication
	$\mid$	$(/.)$	element division
	$\mid$	$(= .)$	element equality
	$\mid$	$(< .)$	element less-than
	$\mid$	$\mathbf{set}$	array index assignment
	$\mid$	$\mathbf{get}$	array indexing
	$\mid$	$\mathbf{share}$	share array
	$\mid$	$\mathbf{unshare}$	unshare array
	$\mid$	$\mathbf{free}$	free arrary
	$\mid$	$\mathbf{array}$	Owl: make array
	$\mid$	$\mathbf{copy}$	Owl: copy array
	$\mid$	$\mathbf{sin}$	Owl: map sine over array
	$\mid$	$\mathbf{hypot}$	Owl: $x_i := \sqrt{x_i^2 + y_i^2}$

	asum	BLAS: $\sum_i \lvert x_i \rvert$
	axpy	BLAS: $x := \alpha x + y$
	dot	BLAS: $x \cdot y$
	rotmg	BLAS: see its docs
	scal	BLAS: $x := \alpha x$
	amax	BLAS: $\operatorname{argmax} i : x_i$
	setM	matrix index assignment
	getM	matrix indexing
	shareM	share matrix
	unshareM	unshare matrix
	freeM	free matrix
	matrix	Owl: make matrix
	copyM	Owl: copy matrix
	copyM_to	Owl: copy matrix onto another
	sizeM	dimension of matrix
	trnsp	transpose matrix
	gemm	BLAS: $C := \alpha A^{T?} B^{T?} + \beta C$
	symm	BLAS: $C := \alpha AB + \beta C$
	posv	BLAS: Cholesky decomp. and solve
	potrs	BLAS: solve with given Cholesky

Θ	::=		fractional capability environment
	|	.	
	|	Θ, fc	

Γ	::=		linear types environment
	|	.	
	|	$\Gamma, x : t$	
	|	Γ, Γ'	

Δ	::=		linear types environment
	|	.	
	|	$\Delta, x : t$	

$\boxed{\Theta \vdash f\ \mathsf{Cap}}$ Valid fractional capabilities

$$\frac{fc \in \Theta}{\Theta \vdash fc\ \mathsf{Cap}}\ \text{WF\_Cap\_Var}$$

$$\frac{}{\Theta \vdash \mathbf{Z}\ \mathsf{Cap}}\ \text{WF\_Cap\_Zero}$$

$$\frac{\Theta \vdash f\ \mathsf{Cap}}{\Theta \vdash \mathbf{S}\ f\ \mathsf{Cap}}\ \text{WF\_Cap\_Succ}$$

$\boxed{\Theta \vdash t\ \mathsf{Type}}$ Valid types

$$\frac{}{\Theta \vdash \mathbf{unit}\ \mathsf{Type}}\ \text{WF\_Type\_Unit}$$

$$\frac{}{\Theta \vdash \mathbf{bool}\ \mathsf{Type}}\ \text{WF\_Type\_Bool}$$

$$\frac{}{\Theta \vdash \mathbf{int}\ \mathsf{Type}}\ \text{WF\_Type\_Int}$$

$$\frac{}{\Theta \vdash \mathbf{elt}\ \mathsf{Type}}\ \textsc{WF_Type_Elt}$$

$$\frac{\Theta \vdash f\ \mathsf{Cap}}{\Theta \vdash f\ \mathbf{arr}\ \mathsf{Type}}\ \textsc{WF_Type_Array}$$

$$\frac{\Theta \vdash t\ \mathsf{Type}}{\Theta \vdash\ !t\ \mathsf{Type}}\ \textsc{WF_Type_Bang}$$

$$\frac{\Theta, fc \vdash t\ \mathsf{Type}}{\Theta \vdash \forall fc.t\ \mathsf{Type}}\ \textsc{WF_Type_Gen}$$

$$\frac{\Theta \vdash t\ \mathsf{Type} \quad \Theta \vdash t'\ \mathsf{Type}}{\Theta \vdash t \otimes t'\ \mathsf{Type}}\ \textsc{WF_Type_Pair}$$

$$\frac{\Theta \vdash t\ \mathsf{Type} \quad \Theta \vdash t'\ \mathsf{Type}}{\Theta \vdash t \multimap t'\ \mathsf{Type}}\ \textsc{WF_Type_Lolly}$$

$\boxed{\Theta; \Delta; \Gamma \vdash e : t}$ Typing rules for expressions

$$\frac{}{\Theta; \Delta; \cdot, x : t \vdash x : t}\ \textsc{Ty_Var_Lin}$$

$$\frac{x : t \in \Delta}{\Theta; \Delta; \cdot \vdash x : t}\ \textsc{Ty_Var}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : t \quad \Theta; \Delta; \Gamma', x : t \vdash e' : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\ x = e\ \mathbf{in}\ e' : t'}\ \textsc{Ty_Let}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash () : \mathbf{unit}}\ \textsc{Ty_Unit_Intro}$$

$$\frac{\Theta; \Delta; \cdot \vdash e : \mathbf{unit} \quad \Theta; \Delta; \Gamma \vdash e' : t}{\Theta; \Delta; \Gamma \vdash \mathbf{let}\ () = e\ \mathbf{in}\ e' : t}\ \textsc{Ty_Unit_Elim}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{true} : !\mathbf{bool}}\ \textsc{Ty_Bool_True}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{false} : !\mathbf{bool}}\ \textsc{Ty_Bool_False}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : \mathbf{bool} \quad \Theta; \Delta; \Gamma' \vdash e_1 : t' \quad \Theta; \Delta; \Gamma' \vdash e_2 : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{if}\ e\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2 : t}\ \textsc{Ty_Bool_Elim}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash k : !\mathbf{int}}\ \textsc{Ty_Int_Intro}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash el : !\mathbf{elt}}\ \textsc{Ty_Elt_Intro}$$

$$\frac{\Theta; \Delta; \cdot \vdash v : t}{\Theta; \Delta; \cdot \vdash \mathbf{Many}\ v : !t}\ \textsc{Ty_Bang_Intro}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : !t \quad \Theta; \Delta, x : t; \Gamma' \vdash e' : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\ \mathbf{Many}\ x = e\ \mathbf{in}\ e' : t'}\ \textsc{Ty_Bang_Elim}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : t \quad \Theta; \Delta; \Gamma' \vdash e' : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash (e, e') : t \otimes t'} \quad \textsc{Ty\_Pair\_Intro}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e_{12} : t_1 \otimes t_2 \quad \Theta; \Delta; \Gamma', a : t_1, b : t_2 \vdash e : t}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\,(a, b) = e_{12}\,\mathbf{in}\,e : t} \quad \textsc{Ty\_Pair\_Elim}$$

$$\frac{\Theta \vdash t' \,\mathsf{Type} \quad \Theta; \Delta; \Gamma, x : t' \vdash e : t}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\,x : t' \to e : t' \multimap t} \quad \textsc{Ty\_Lambda}$$

$$\frac{\Theta; \Delta; \Gamma \vdash e : t' \multimap t \quad \Theta; \Delta; \Gamma' \vdash e' : t'}{\Theta; \Delta; \Gamma, \Gamma' \vdash e\ e' : t} \quad \textsc{Ty\_App}$$

$$\frac{\Theta, fc; \Delta; \Gamma \vdash e : t}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\,fc \to e : \forall\, fc.t} \quad \textsc{Ty\_Gen}$$

$$\frac{\Theta \vdash f \,\mathsf{Cap} \quad \Theta; \Delta; \Gamma \vdash e : \forall\, fc.t}{\Theta; \Delta; \Gamma \vdash e[f] : t\{f/fc\}} \quad \textsc{Ty\_Spc}$$

$$\frac{\Theta; \Delta, g : t \multimap t'; \cdot, x : t \vdash e : t'}{\Theta; \Delta; \cdot \vdash \mathbf{fix}\,(g, x : t, e : t') : !(t \multimap t')} \quad \textsc{Ty\_Fix}$$

```
Definition rules:         19 good    0 bad
Definition rule clauses: 43 good    0 bad
```